中国科学 E辑：技术科学 2009 年 第 39 卷 第 4 期：642 ~ 647
www.scichina.com tech.scichina.com
《中国科学》杂志社
SCIENCE IN CHINA PRESS

采用内嵌边界方法模拟流体中颗粒运动的并行算法

王则力，樊建人*，罗坤

浙江大学能源清洁利用国家重点实验室，杭州 310027

* E-mail: fanjr@zju.edu.cn

收稿日期: 2008-03-05; 接受日期: 2008-05-12

国家自然科学基金资助项目(批准号: 50736006, 50506027, 50776080)

摘要 提出了一种采用内嵌边界方法模拟流体中颗粒运动的并行算法. 当计算区域分区并行时，该算法能够很好的处理颗粒穿过亚区域边界时颗粒与流体之间的耦合问题. 模拟计算了两维颗粒在密闭方形区域中自由沉降问题来验证该方法的有效性.

关键词
并行计算
内嵌边界方法
颗粒运动

用于模拟虚拟边界与流体之间相互作用的内嵌边界方法[1~10]最早是由Peskin[1]提出并用以模拟研究心脏内血液的运动. 随着研究的深入，内嵌边界方法被用来模拟研究流体与固定边界之间的相互作用[1,2,4,5,7]以及流体与振动边界之间的相互作用[4]. 近年来，内嵌边界方法已经被用于研究多相流中在流体的作用下颗粒的运动[9,11~14].

为了计算流体中固体边界与流体之间的相互作用，Goldstein等人[15]提出了一种“反馈力”算法. 在反馈力的计算中，通过迭代求解使得内嵌边界上的速度满足无滑移便捷条件，从而获得颗粒壁面与流体之间的相互作用力. Saiki和Biringen[4]采用这种“反馈力”算法结合内嵌边界方法模拟了流体绕过圆环. Fadlun等人[5]在简单内嵌边界方法的基础上提出了一种“直接力”算法用于计算流体与固体壁面之间的相互作用力. Wang等人[14]提出了“多重直接力”的算法并结合内嵌边界方法研究了多相流系统中固体颗粒的运动. 同时成功的模拟并观察到多相流中的一些微观现象，例如：当2个颗粒互相靠近时发生吸引-碰撞-翻转现象；2 个颗粒碰撞时颗粒之间垂直于颗粒碰撞方向上会产生射流.

在多相流系统中，当采用内嵌边界方法模拟流体中颗粒的运动时，由于计算量较大[9]，因而采用并行计算方法进行模拟显得十分必要. Uhlmann[16]采用了主-从算法来并行求解颗粒的运动. 在这种算法中，颗粒“主”处理器用于处理所有颗粒的通常计算，而一系列颗粒“从”处理器用于计算特定的颗粒. 这种颗粒的并行算法避免了颗粒穿越亚计算区域时所产生的问题.

本文提出了一种颗粒的并行计算方法，该方法能够很好的处理采用内嵌边界方法模拟流体中颗粒运动时颗粒穿过亚计算区域边界问题，同时该方法能够很容易的进行编程. 本文第 1 节简要的介绍采用内嵌边界方法模拟流体中颗粒运动的数学方法. 第2节为采用内嵌边界方法模拟流体中颗粒运动的并行算法. 我们采用该方法模拟了二维单个颗粒在密闭区域内自由沉降问题(第 3 节)并对并行效率进行了分析(第 4 节). 最后，第 5 节为结论.

1 数值方法

在整个计算区域Ω内，粘性不可压缩流体的控制

方程为

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla P + \frac{1}{Re}\nabla^2 \boldsymbol{u} + \boldsymbol{f}, \tag{2}$$

其中, $\boldsymbol{u}$ 为流体速度, P 为流体压力, $Re = \dfrac{\rho \cdot U \cdot L}{\mu}$ 为 Reynolds 数(ρ 为流体密度, U 为特征速度, L 为特征长度, μ 为流体粘度). 方程(2)中的外力 $\boldsymbol{f}$ 为流体与颗粒内嵌边界之间的相互作用力, 其计算公式如下

$$\boldsymbol{f}(x) = \int_{\Omega} \boldsymbol{F}_k(\boldsymbol{x}_k) \cdot \delta(\boldsymbol{x} - \boldsymbol{x}_k) \mathrm{d}\boldsymbol{x}_k, \tag{3}$$

其中, $\delta(\boldsymbol{x} - \boldsymbol{x}_k)$ 是 Dirac 函数, $\boldsymbol{x}_k$ 为内嵌边界上拉格朗日点所在位置, $\boldsymbol{x}$ 为欧拉网格所在位置, $\boldsymbol{F}_k(\boldsymbol{x}_k)$ 为内嵌边界拉格朗日点 $\boldsymbol{x}_k$ 上受到的作用力.

Dirac函数是连接内嵌边界上拉格朗日点和欧拉计算网格点的桥梁. 为了进行数值计算, Dirac函数被离散为[8]

$$\delta_h(\boldsymbol{x} - \boldsymbol{x}_k) = \frac{1}{h^2} d_h\left(\frac{x - x_k}{h}\right) \cdot d_h\left(\frac{y - y_k}{h}\right), \tag{4}$$

$$d_h(r) = \begin{cases} \dfrac{1}{8}(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}), & 0 \leqslant |r| < 1, \\ \dfrac{1}{8}(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}), & 1 \leqslant |r| < 2, \\ 0, & 2 \leqslant |r|, \end{cases} \tag{5}$$

其中, $\boldsymbol{x} = (x, y)$, $\boldsymbol{x}_k = (x_k, y_k)$, h 为欧拉计算网格长度.

我们采用Wang等人[14]提出的多重直接力算法来耦合颗粒与流体, 从而使得颗粒边界处的流体速度满足无滑移边界条件. 内嵌边界拉格朗日点上受到的流体与颗粒之间的相互作用力可以通过(6)式获得, 即采用NF次直接力作用于内嵌边界构成了多重直接力. 颗粒受到流体的水力学作用力和力矩可分别通过(7)和(8)式获得.

$$\boldsymbol{F}_k(\boldsymbol{x}_k) = \sum_{i=1}^{NF} \boldsymbol{F}_k^i(\boldsymbol{x}_k), \tag{6}$$

$$\boldsymbol{F} = -\int_1^N \boldsymbol{F}_k(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{s} = -\int_{\Omega} \boldsymbol{f}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = -\sum_{\boldsymbol{x} \in \Omega} \boldsymbol{f}(\boldsymbol{x}) h^2, \tag{7}$$

$$\begin{aligned} \boldsymbol{T} &= -\int_1^N (\boldsymbol{x}_k - \boldsymbol{x}_c) \times \boldsymbol{F}_k(\boldsymbol{x}_k)\mathrm{d}\boldsymbol{s} \\ &= -\int_{\Omega} (\boldsymbol{x} - \boldsymbol{x}_c) \times \boldsymbol{f}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \\ &= -\sum_{\boldsymbol{x} \in \Omega} (\boldsymbol{x} - \boldsymbol{x}_c) \times \boldsymbol{f}(\boldsymbol{x}) h^2, \end{aligned} \tag{8}$$

其中, N 为颗粒内嵌边界上拉格朗日点的数目.

颗粒的运动可以通过求解牛顿运动方程获得

$$M\frac{\mathrm{d}\boldsymbol{U}}{\mathrm{d}t} = M\boldsymbol{g} + \boldsymbol{F} + \boldsymbol{F}', \tag{9}$$

$$\frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t} = \boldsymbol{U}, \tag{10}$$

$$\boldsymbol{I}\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} = \boldsymbol{T}, \tag{11}$$

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \boldsymbol{\omega}, \tag{12}$$

其中, M 为颗粒质量, $\boldsymbol{U}$ 为颗粒速度, $\boldsymbol{g}$ 为重力加速度, $\boldsymbol{X}$ 为颗粒质心所在位置, I 为颗粒转动惯量, $\boldsymbol{\omega}$ 为颗粒旋转角速度, $\boldsymbol{\theta}$ 为颗粒旋转角度, $\boldsymbol{F}'$ 为颗粒之间或颗粒与壁面之间相互碰撞时的作用力.

2 并行计算方法

采用内嵌边界方法模拟流体中颗粒的运动可分为 2 步进行计算. 第 1 步是流场的求解, 流场的求解我们在欧拉网格上直接求解 Navier-Stokes 方程获得流场解, 此时的计算效率取决于欧拉计算网格量. 第 2 步是颗粒与流场之间的耦合, 颗粒与流场之间的耦合我们采用内嵌边界方法结合多重直接力算法获得. 在第 2 步中, 当采用区域分块算法时, 计算颗粒运动通过亚区域边界时颗粒与流体之间的耦合是一个耗时的问题. 因此采用内嵌边界方法模拟流体中颗粒运动的并行算法可以分为 2 部分, 即并行求解流场和并行计算颗粒运动.

2.1 流场并行算法

对于一个方形计算区域Ω(图 1(a))可分为 N 块尺度相等的亚区域Ω_i (i=1~N).(图 1(b), 例如 N=9). 每块亚区域采用一个CPU进行求解. 为了保证在每个计算时间步长内, 亚区域边界上的数据保持连续性, 我们采用了3层欧拉网格重叠的方法来进行数据之间的耦合, 如图1(c)所示. 亚区域边界上网格点上的数据可以通过直接求解其相邻亚区域内的控制方程并通过数据传递获得. 亚区域边界倒数第2层网格上的

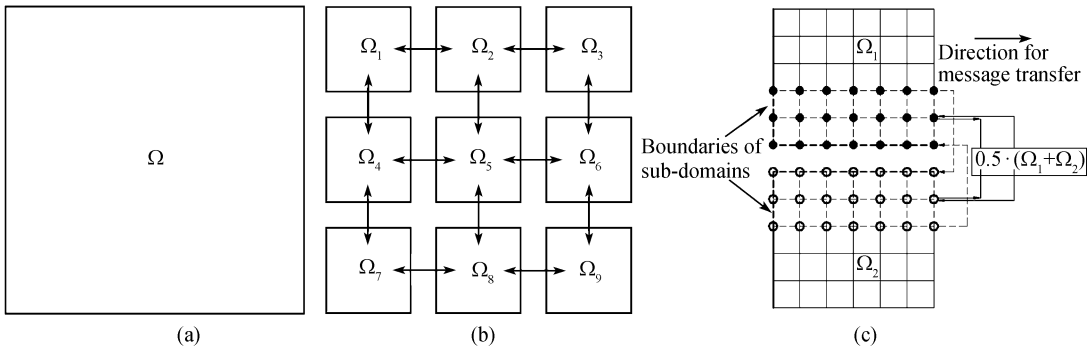

图 1　计算区域分解和亚区域间数据传递方向

数据可以同时通过求解2个相连亚区域内的控制方程获得. 为了保证亚区域边界上的连续性, 亚区域边界倒数第 2 层网格上的数据值为相连 2 个区域内在该处数据的平均值. 亚区域之间的数据传递方向如图 1(b)所示, 对于二维计算, 一个亚区域的数据最大传递方向为 4 个方向. 这样的区域分解算法能够很好的保证各个CPU上计算负载的平衡. 同时, 亚区域之间的即时通讯能够保证解的光滑型和收敛性[17].

2.2　颗粒运动并行算法

当采用内嵌边界方法计算流体中颗粒的运动时, 流体与固体颗粒之间的耦合通过内嵌边界上的拉格朗日点与欧拉网格节点之间的内/外插值来实现. 离散Dirac函数在耦合中起着关键作用. 本文中所采用的离散Dirac函数为(4)和(5) 式, 其影响区域为 2 个欧拉网格长度如图 2 所示. 当颗粒正通过亚区域边界时, 一个亚区域内欧拉网格上的参数可能会受到其相邻亚区域内拉格朗日点的影响, 因而会增加 2 个亚区域之间的数据传递的复杂性[16]. 为了避免这个问题, 我们采用流场亚区域边界上 3 层网格重叠方法来耦合颗粒与流场之间的相互作用. 如图 3 所示, 拉格朗日点A在亚区域Ω_1 中距离边界两个欧拉网格长度距离亚区域Ω_2 边界一个欧拉网格长度处, 通过离散Dirac函数作用后, 点A的影响区域为其周围的 9 个欧拉节点, 并标记为×. 亚区域Ω_1包含了所有标记的欧拉网格节点, 而亚区域Ω_2只包含了一部分标记的欧拉网格节点. 因此拉格朗日点与欧拉网格节点之间的内/外插值过程全部在亚区域Ω_1中完成, 不需要在亚区域Ω_2中进行计算. 当颗粒进行水平运动, 其边界上拉格朗日点由 A 运动到了 B. 点 B 的影响区域既包含在亚区域Ω_1 中, 又包含在亚区域Ω_2 中. 在这种状态下, 拉格朗日点与欧拉网格节点之间的内/外

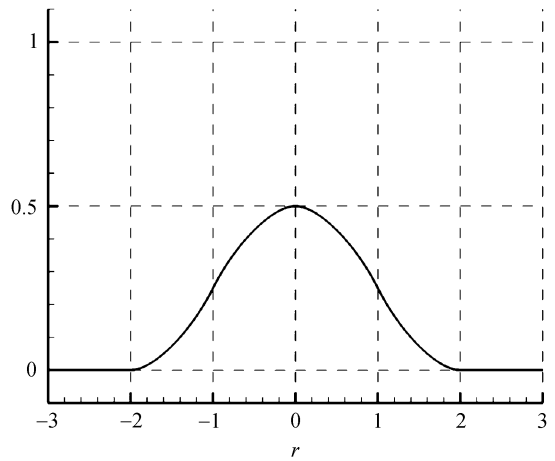

图 2　离散 Dirac 函数

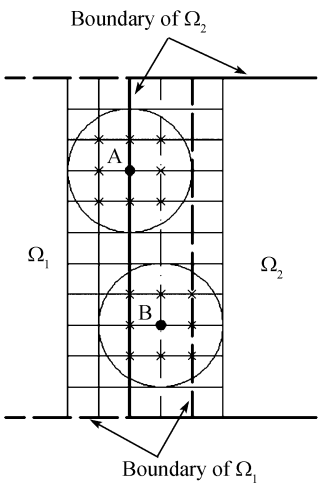

图 3　三层网格重叠示意图

插值过程可以同时在 2 个亚区域内进行, 因而不需要进行数据传递. 当颗粒边界上拉格朗日点位于B点的右侧时, 拉格朗日点与欧拉网格节点之间的内/外插值过程可全部在亚区域Ω_2 中完成. 同时为了节约计算资源, 在亚区域Ω_1 中存储的关于拉格朗日点的信息可以清除.

采用这样的算法计算颗粒运动通过亚区域边界时, 由于不需要在亚区域间进行有关拉格朗日点上数据的传递, 因而具有较高的计算效率并且十分易于编程. 同时, 由于离散 Dirac 函数的影响区域为 2 倍欧拉网格长度, 因而亚区域边界上重叠 3 层欧拉网格十分必要. 对于其他形式的离散 Dirac 函数, 如果其影响区域为 $N_{\inf}$ 倍欧拉网格长度, 那么亚区域边界上欧拉网格重叠数为 $N_{\inf}+1$.

对于多相流系统, 采用上述方法进行颗粒间的秉性计算时, 颗粒计算附载的平衡取决于颗粒在各个亚计算区域内的分布. 由于颗粒具有有限体积, 因而每个亚区域内所能容纳的颗粒数目是有限的. 而每个亚区域内的最大的计算量取决于内迁边界上拉格朗日点的树木以及欧拉网格的数目.

3 数值验证

我们采用内嵌边界方法以及本文提出的并行算法数值模拟了密闭区域内单颗粒的自由沉降. 计算区域大小为 $\Omega=(0,2)\times(0,6)$, 颗粒直径为 $D_P=0.25$, 颗粒密度为 $\rho_P=1.25$. 颗粒初始位置为(1, 4). 流体密度为 $\rho_f=1.00$, 流体粘度为 $\mu=0.1$. 计算欧拉网格长度为颗粒直径的1/64, 即 h=1/256. 欧拉网格计算节点数为 788481, 内迁边界上拉格朗日点数目为 202. 计算求解的时间步长为 5×10^{-5}. 为了便于分析并行效率, 整个计算区域分解为 2, 3, 4 和 8 个区域并与没有分区的结果进行比较. 在计算中每个CPU用于计算一个亚区域, 区域之间的数据传递采用MPI实现[18].

最大雷诺数定义为

$$Re_{\max}=\max[Re(t)]=\max\left[\frac{\rho_P\sqrt{U_P(t)^2+V_P(t)^2}\cdot D_P}{\mu}\right], \quad (13)$$

其中, $U_P(t)$ 和 $V_P(t)$ 为颗粒速度在水平和竖直方向上的分量.

图 4 为颗粒自由沉降时, 颗粒位置(左图)和颗粒沉降速度(右图)随时间变化关系. 当采用不同数目的CPU进行计算时, 颗粒的沉降位置和颗粒沉降速度随时间变化曲线完全重叠, 这表明采用不同的分区并行计算能够获得相同的计算结果并且颗粒能够光滑地通过各个亚区域边界. 因此采用 3 层欧拉网格重叠的区域分解算法能够很好地处理并行计算中颗粒穿过亚区域边界问题并能获得光滑解.

在各种分区求解中, 颗粒沉降时最大雷诺数均为 17.32, 这与Glowinski[19]的数值计算结果 17.31十分吻合. 在t=0.65 时流场结构和颗粒位置如图 5(a)~(e)所示. 与没有采用分区求解得到的结果相比(图 5(a)), 采用不同的分区(图5(b)~(e))模拟单颗粒在密闭区域中自由沉降时能够获得一致的解. 因此, 第 2 节中提出的采用内嵌边界方法模拟流体中颗粒运动的并行算法能够有效地用于模拟多相流运动.

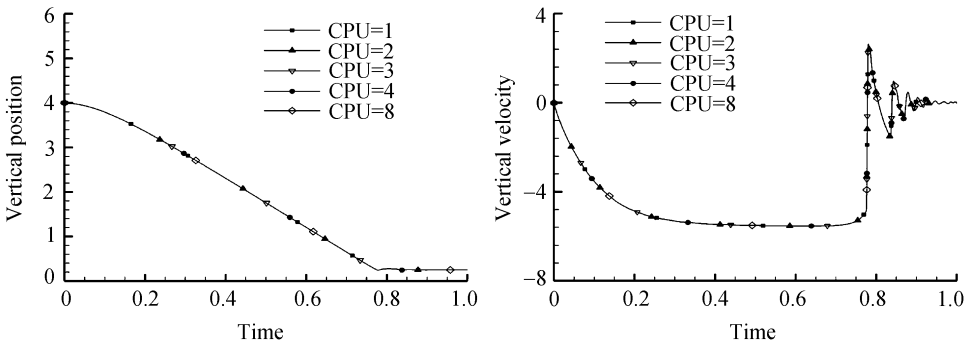

图 4 颗粒位置(左图)和颗粒沉降速度(右图)随时间变化关系

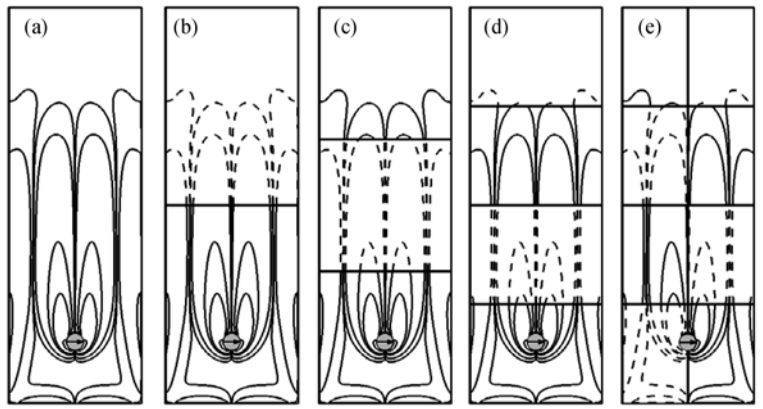

图 5　在 t=0.65 时颗粒位置和流场涡结构图

(a) 为不采用并行的结果; (b)~(e)分别为 2,3,4,8 个亚区并行计算时的结果

4　并行效率分析

采用不同区域分块并行计算得到的颗粒在密闭方形区域内自由沉降的结果如第 3 节所述. 为了比较本文所提出的并行算法的计算效率, 我们对采用不同个数 CPU 时的计算耗时、加速比、并行效率进行比较, 如表 1 所示.

表 1　单颗自由沉降时每个时间步长内的并行计算效率

CPU 数目	耗时/s	加速比	并行效率
1	7.66	1.00	1.00
2	4.08	1.88	0.94
3	3.13	2.45	0.82
4	2.54	3.02	0.76
8	1.66	4.61	0.58

当采用 2 个CPU进行计算时, 并行效率较高为 0.94. 随着CPU个数的增加, 并行效率降低. 对于固定计算量的工作, CPU数目的增加使得通讯耗时的增加, 同时CPU数目的增加使得每个CPU的计算量减小, 计算耗时减小, 因而使得并行效率的降低[20]. 而在每个CPU上的计算量固定, 而计算总量随CPU个数增加的情况下, 本文提出的用于全尺度计算颗粒运动的并行计算方法则具有较高的并行效率和加速比.

5　结论

本文提出了一种采用内嵌边界方法模拟流体中颗粒运动的并行算法, 并采用该方法模拟计算了两维颗粒在密闭方形区域中自由沉降问题来验证方法的有效性. 该方法的优点可总结如下.

1) 能很好地处理当颗粒穿过亚区域边界时颗粒与流体之间的耦合.

2) 对于每个 CPU, 最大计算量是固定的.

3) 具有较高的并行计算的效率和加速比.

本文提出的采用内嵌边界方法模拟流体中颗粒运动的并行算法是基于两维的计算. 由于采用了基于笛卡儿均匀网格的内嵌边界方法来模拟计算流体中颗粒的运动, 因而本文提出的并行算法能够很容易地、直接地扩展到三维颗粒与流体之间相互作用的计算. 我们进一步的工作是采用内嵌边界方法结合多重直接力算法以及并行计算方法来全尺度研究三维气固多相流.

参考文献

1　Peskin C S. Flow patterns around heart valves: A numerical method. J Comput Phys, 1972, 10(2): 252—271

2　Peskin C S. Numerical analysis of blood flow in the heart. J Comput Phys, 1977, 25(3): 220—252

3　Beyer R P, Leveque R J. Analysis of a one-dimensional model for the immersed boundary method. SIAM J Numer Anal, 1992, 29(2): 332—364[DOI]

4　Saiki E M, Biringen S. Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. J Comput Phys,

1996, 123(2): 450—465 [DOI]
5 Fadlun E A, Verzicco R, Orlandi P, et al. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J Comput Phys, 2000, 161(1): 35—60[DOI]
6 Peskin C S. The immersed boundary method. Acta Numer, 2002, 11: 479—517 [DOI]
7 Silva A L F L E, Silveira-Neto A, Damasceno J J R. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. J Comput Phys, 2003, 189(2): 351—370[DOI]
8 Griffith B E, Peskin C S. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. J Comput Phys, 2005, 208(1): 75—105[DOI]
9 Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. J Comput Phys, 2006, 209(2): 448—476[DOI]
10 Zhang N, Zheng Z C. An improved direct-forcing immersed-boundary method for finite difference applications. J Comput Phys, 2007, 221(1): 250—268[DOI]
11 Niu X D, Shu C, Chew Y T, et al. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. Phys Lett A, 2006, 354(3): 173—182[DOI]
12 Luo K, Wang Z L, Fan J R. A modified immersed boundary method for simulations of fluid-particle interactions. Comput Methods Appl Mech Engrg, 2007, 197: 36—46[DOI]
13 Luo K, Wang Z L, Fan J R, et al. Full-scale solutions to particle-laden flows: Multiphase forcing and immersed boundary method. Phys Rev E, 2007, 76: 066709
14 Wang Z L, Fan J R, Luo K. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. Int J Multiphase Flow, 2008, 34(3): 283—302[DOI]
15 Goldstein D, Handler R, Sirovich L. Modeling a no-slip boundary with an external force field. J Comput Phys, 1993, 105(2): 354—366[DOI]
16 Uhlmann M. Simulation of particulate flows on multi-processor maschines with distributed memory. CIEMAT Technical Repot No. 1039, Madrid, Spain, ISSN-1135-9420, 2003
17 Taneda S, Hongji H. Unsteady flow past a flat normal to the direction of motion. J Phys Soc Japan, 1971, 30(1): 262—272
18 Ashton D, Gropp W, Lusk E. Installation and user's guide to MPICH, a portable implementation of MPI version 1.2.5, Mathematics and computer science division, ANL/MCS-TM-ANL-01/x Revx, http://www-unix.mcs.anl.gov/mpi/mpich1/docs.html
19 Glowinski R, Pan T W, Hesla T I, et al. A distributed Lagrangian multiplier/fictitious domain method for particulate flow. Int J Multiphase Flow, 1999, 25(5): 755—794[DOI]
20 Amdahl G M. Validity of single-processors approach to achieving large-scale computing capacity. In: Proceedings AFIPS Conference. Reston, VA: AFIPS Press, 1967. 483—486